

Date⇒ 05-01-2021

Module⇒ Backend

Lecture By⇒ Akash Handa

Subject ⇒

IN PREVIOUS LECTURE (QUICK RECAP) Date-04/02/2021	In Today's Lecture (Overview)
What is Crud Create Read Update Delete What is Cors What is Body parser What is Bson and What is the difference between Bson And Json How To Connect Nodejs With Mongodb	Creating simple Crud App using Nodejs And Mongodb

Creating simple Crud App using Nodejs And Mongodb

App.js

```
const express = require('express');
const app = express();
const port = process.env.PORT || 9700;
const mongo = require('mongodb');
const MongoClient = mongo.MongoClient
const bodyParser = require('body-parser');
const cors = require('cors');
const mongourl = `mongodb://localhost:27017`;
let dbobj;
let col_name="users";

//middleware
//cross origin resource sharing
app.use(cors())

//parse data for post call
```

```
app.use(bodyParser.urlencoded({extended:true}))
app.use(bodyParser.json())

//ejs
app.use(express.static(__dirname+'/public'));
app.set('views','./src/views');
app.set('view engine','ejs');

//health Check
app.get('/health',(req,res) => {
    res.status(200).send("Health Ok")
})

//load home page
app.get('/',(req,res) => {
    dbobj.collection(col_name).find({isActive:true}).toArray((err,result)
=> {
        if(err) throw err;
        res.render('index',{data:result})
    })
})

app.get('/new',(req,res) => {
    res.render('admin')
})

//postUser
app.post('/adduser',(req,res)=>{
    const data = {
        "name": req.body.name,
        "city": req.body.city,
        "phone": req.body.phone,
        "isActive": req.body.isActive?req.body.isActive:true,
        "role": req.body.role?req.body.role:'User',
        "email": req.body.email
    }
    dbobj.collection(col_name).insert(data,(err,result) => {
        if(err) throw err;
        //res.status(200).send("Data Added")
        res.redirect('/')
    })
})
```

```

    });
  });

//getUser
app.get('/users', (req, res) => {
  var query = {}
  if(req.query.city && req.query.role){
    query={city:req.query.city,role:req.query.role,isActive:true}
  }
  else if(req.query.city){
    query={city:req.query.city,isActive:true}
  }else if(req.query.role){
    query={role:req.query.role,isActive:true}
  }else{
    query={isActive:true}
  }
  dbobj.collection(col_name).find(query).toArray((err,result) => {
    if(err) throw err;
    res.send(result)
  })
})

//UserDetails
app.get('/user/:id', (req, res) => {
  var id = mongo.ObjectID(req.params.id)
  var query = {}
  query={_id:id}
  dbobj.collection(col_name).findOne(query, (err,result) => {
    if(err) throw err;
    res.send(result)
  })
})

//updateUser
app.put('/editUser', (req, res) => {
  let Id = req.body._id;
  dbobj.collection(col_name).update(
    {_id:mongo.ObjectID(Id)},
    {
      $set:{

```

```

        name:req.body.name,
        email:req.body.email,
        city: req.body.city,
        phone: req.body.phone,
        isActive:req.body.isActive?true:false,
        role: req.body.role,
    }
}, (err, result) => {
    if(err) throw err;
    res.send("Data Updated")
})
)
}))

//hardDeleteUser
app.delete('/deleteUser', (req, res) => {
    let Id = mongo.ObjectId(req.body._id);
    dbobj.collection(col_name).remove({_id:Id}, (err, result) => {
        if(err) throw err;
        res.send("Data Deleted")
    })
});

//deactivate User(soft Delete)
app.put('/deactivateUser', (req, res)=>{
    let Id = mongo.ObjectId(req.body._id);
    dbobj.collection(col_name).update(
        {_id:Id},
        {
            $set:{
                isActive:false
            }
        }, (err, result)=> {
            if(err) throw err;
            res.send("User Deactivated")
        }
    )
});

//Activate User(soft Delete)

```

```

app.put('/activateUser', (req, res) => {
  let Id = mongo.ObjectId(req.body._id);
  dbobj.collection(col_name).update(
    {_id: Id},
    {
      $set: {
        isActive: true
      }
    }, (err, result) => {
      if(err) throw err;
      res.send("User Activated")
    }
  )
})

MongoClient.connect(mongourl, (err, connection) => {
  if(err) console.log(err);
  dbobj = connection.db('aryabhat');
  app.listen(port, (err) => {
    console.log(`Server is running on port ${port}`)
  })
})

```

Index.ejs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Add new Data</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.cs
s">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></s
cript>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.mi
n.js"></script>

```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js">
</script>

<!--DataTable-->
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.css">
<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.js"></script>
</head>
<body>

<div class="container">
  <br>
  <nav class="navbar navbar-expand-md bg-dark navbar-dark
fixed-top">
    <!-- Brand -->
    <a class="navbar-brand" href="#">Developer Funnel</a>

    <!-- Toggler/collapsible Button -->
    <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>

    <!-- Navbar links -->
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/new">Add new</a>
        </li>
      </ul>
    </div>
  </nav>
  <br/><br/>
  <div class="table-responsive">
```

```

        <table class="table table-hover table-striped"
id="myTable">
    <thead class="thead-dark">
        <tr>
            <th>Name</th>
            <th>City</th>
            <th>Phone</th>
            <th>Email</th>
            <th>Role</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody>
        <%for (i=0;i<data.length;i++) {%>
            <tr>
                <td><%=data[i].name%></td>
                <td><%=data[i].city%></td>
                <td><%=data[i].phone%></td>
                <td><%=data[i].email%></td>
                <td><%=data[i].role%></td>
                <td>
                    <button type="button" class="edit btn
btn-warning"
                    data-toggle="modal"
data-target="#myModal"
                    id="<%=data[i]._id%>">Edit</button>
                    <button type="button" class="btn
btn-danger" id="<%=data[i]._id%>">Delete</button>
                </td>
            </tr>
        <%}%>
    </tbody>
</table>
<div class="modal" id="myModal">
    <div class="modal-dialog">
        <div class="modal-content">

            <!-- Modal Header -->
            <div class="modal-header">
                <h4 class="modal-title">Modal Heading</h4>

```

```

        <button type="button" class="close"
data-dismiss="modal">&times;</button>
    </div>

    <!-- Modal body -->
    <div class="modal-body">
        <form action="/editUser" method="POST"
id="update_user">
            <input type="text" readonly
class="form-control" id="update_id" name="_id">
            <div class="form-group">
                <label for="name">Name:</label>
                <input type="text"
class="form-control" id="update_name" placeholder="Enter First name"
name="name" required>
            </div>
            <div class="form-group">
                <label for="city">City:</label>
                <input type="text"
class="form-control" id="update_city" placeholder="Enter City"
name="city">
            </div>
            <div class="form-group">
                <label for="phone">Phone:</label>
                <input type="phone"
class="form-control" id="update_phone" placeholder="Enter phone"
name="phone">
            </div>
            <div class="form-group">
                <label for="email">Email:</label>
                <input type="email"
class="form-control" id="update_email" placeholder="Enter Email"
name="email">
            </div>
            <div class="form-group">
                <label for="role">Role:</label>
                <input type="role"
class="form-control" id="update_role" name="role" readonly>
            </div>
            <div class="form-group">

```



```
</label>
for="role">IsActive:</label>

<input type="role"
class="form-control" id="update_active" name="isActive" readonly>
</div>

</form>

</div>

<!-- Modal footer -->
<div class="modal-footer">
    <button type="button" class="btn btn-danger" data-dismiss="modal">Close</button>
    <button type="submit" id="update_table" class="btn btn-primary">update</button>
</div>

</div>
</div>
</div>
</div>
</script>
$(document).ready(function() {
    $('#myTable').DataTable();
})
$('.edit').click(function(){
    id=this.id;
    console.log(id)
$.ajax({
    type:'get',
    url:`/user/${id}`,
    success:function(data){
        //console.log(data)
        $("#update_id").attr("value",data._id);
        $("#update_name").attr("value", data.name);
        $("#update_city").attr("value", data.city);
        $("#update_phone").attr("value", data.phone);
        $("#update_role").attr("value", data.role);
```

```
        $("#update_email").attr("value", data.email);
        $("#update_active").attr("value", data.isActive);
    }
    })
    })
    $(function(){
        $('#update_table').on('click',function(){
            var data = $('#update_user').serialize()
            console.log(data)
            $.ajax({
                url:'/editUser',
                type:'PUT',
                data:data,
                success:function(data){
                    console.log("Success")
                    window.location.reload()
                },
                error:function(){
                    alert("Invalid Action")
                }
            })
        })
    })
</script>
</body>
</html>
```